UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/779,855	02/17/2004	Gosagan Padmanabhan	852663.405	6239

38106 7590 10/30/2007
SEED INTELLECTUAL PROPERTY LAW GROUP PLLC
701 FIFTH AVENUE, SUITE 5400
SEATTLE, WA 98104-7092

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	
MAIL DATE	**DELIVERY MODE**
10/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/779,855	PADMANABHAN ET AL.	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 September 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *3-8,11-19 and 25-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *3-8,11-19 and 25-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 3-9, 11-19, and 25-30 have been considered. Claims 25-30 added as per Applicant's request. Claims 1-2, 9-10 and 20-24 canceled as per Applicant's request. Claims 3, 5-8, and 11-17 amended as per Applicant's request.

Claim Objections

2. Claim 29 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Claim 28 says that the flag is asserted if the extended portion of the program counter is not null, and Claim 29 says that the flag is asserted if the extended portion of the program counter contains an extended address value. However, given that as a standard, a program counter always contains an address, Examiner does not see how these two limitations are different from one another, and as a result, does not see how Claim 29 further limits Claim 28.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 3-9, 11-19, and 25-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicants admitted prior art (herein Padmanabhan), in view of Christie (International Publication WO 02/13005), further in view of Pilat et al. (USPN 4,448,173)

5. As per Claim 3, Padmanabhan teaches: A microprocessor (Page 1, Line 12) comprising:

a memory array having a stack for saving contextual data (Page 1, Lines 16-17); and

a central processing unit coupled to the memory array (Page 1, Line 13), the central processing unit having registers containing contextual data (Page 1, Lines 22-23) and a stack pointer (Page 2 Lines 19-23) and being arranged for saving contextual data upon a switch from a first to a second program (Page 2, Lines 10-23), but fails to teach:

in a variable number of registers that varies according to the value of at least one flag stored in a register to be saved,

wherein the central processing unit is arranged for changing the value of the at least one flag according to the content of an extended addressing register of a program counter of the central processing unit before saving contextual data contained in a variable number of registers that varies according to the value of the at least one flag.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers, allowing more operands to be stored in fast

memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used for vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teaches of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control

register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie and Padmanabhan alone.

Furthermore, regarding the limitation of the flag changing according to the value of an extended addressing register of a program counter, while Christie uses a signal to actually modify the flag, Examiner believes there is a basis for it being "according to the program counter". If the extended portion of the program counter (which is inherent in computers) is invalid, the machine clearly cannot possibly be in the extended mode, and if it does contain valid data, than the machine must be in extended mode, thus the extended addressing portion of the program counter is an indication of what the flag needs to be, and even if Christie uses a signal to switch the flag, the extended portion of the program counter could also do it, based on if it contains valid data or not.

6. As per Claim 4, Christie teaches: The microprocessor according to claim 3 wherein the central processing unit is arranged for:

when the content of the extended addressing register is equal to 0, saving all the registers of the central processing unit containing contextual data, except for the extended addressing register,

when the content of the extended addressing register is not equal to 0, saving all the registers of the central processing unit containing contextual data, including the extended addressing register.

As explained in the rejections for Claim 3, if the extended addressing registers have data, then they have been enabled as taught by Christie. Given the combination with the other references, in which the extended registers are backed up only if they are enabled, then in the case when the content is not equal to zero, saving all of the registers including the extended addressing register would occur as explained above. While the specific teachings of what happens when the content is equal to 0 is not taught by the references (saving all but a single register), it is not required that the value would ever be zero, it is very possible that the value would always be non-zero, thus the references read on the claim in the situation that the content of the extended addressing register is always the same, and in this particular example, non-zero.

7. As per Claim 5, Pilat teaches: The microprocessor according to claim 3 wherein the central processing unit is arranged for performing a test on the value of the at least one flag so as to determine the number of registers to be saved (Column 5, Lines 5-13, the mode of operation must be tested to determine how much data to save).

8. As per Claim 6, Pilat teaches: The microprocessor according to claim 3 wherein the central processing unit is arranged for, upon the return to the first program:

restoring the register containing the at least one flag at a first time (Column 4, Lines 38-39, a return must return all previously saved states); and

restoring contextual data contained in a variable number of registers that varies according to the value of the at least one flag present in the restored register at a second time subsequent to the first time (Column 5, Lines 29-34).

9. As per Claim 7, Christie teaches: The microprocessor according to claim 3 wherein the central processing unit is arranged for saving the register containing the at least one flag last (Page 11, Lines 14-20).

10. As per Claim 8, Christie teaches: The microprocessor according to claim 3 wherein the at least one flag comprises at least one bit of a register containing condition code flags (Page 3, Lines 15-19).

11. As per Claim 11, Padmanabhan teaches: A method for managing the stack of a microprocessor having a central processing unit (Page 1, Line 13) and a memory array (Page 1, Lines 16-17), the central processing unit having registers containing contextual data (Page 1, Lines 22-23) and a stack pointer (Page 2, Lines 19-23), the stack being a zone of the memory array dedicated to saving contextual data upon a switch from a first to a second program (Page 2, Lines 10-23), but fails to teach:

saving contextual data contained in a variable number of registers that varies according to the value of at least one flag stored in a register to be saved; and

changing the value of the at least one flag according to the content of an extended addressing register of a program counter of the central processing unit before saving the contextual data.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers , allowing more operands to be stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used for vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume,

thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teaches of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie alone.

Furthermore, regarding the limitation of the flag changing according to the value of an extended addressing register of a program counter, while Christie uses a signal to actually modify the flag, Examiner believes there is a basis for it being "according to the program counter". If the extended portion of the program counter (which is inherent in computers) is invalid, the machine clearly cannot possibly be in the extended mode, and

if it does contain valid data, than the machine must be in extended mode, thus the extended addressing portion of the program counter is an indication of what the flag needs to be, and even if Christie uses a signal to switch the flag, the extended portion of the program counter could also do it, based on if it contains valid data or not.

12. As per Claim 12, Christie teaches: The method according to claim 11, comprising the following steps:

when the content of the extended addressing register is equal to 0, saving all the registers of the central processing unit containing contextual data, except for the extended addressing register; and

when the content of the extended addressing register is not equal to 0, saving all the registers of the central processing unit containing contextual data, including the extended addressing register.

As explained in the rejections for Claim 11, if the extended addressing registers have data, then they have been enabled as taught by Christie. Given the combination with the other references, in which the extended registers are backed up only if they are enabled, then in the case when the content is not equal to zero, saving all of the registers including the extended addressing register would occur as explained above. While the specific teachings of what happens when the content is equal to 0 is not taught by the references (saving all but a single register), it is not required that the value would ever be zero, it is very possible that the value would always be non-zero, thus the

references read on the claim in the situation that the content of the extended addressing register is always the same, and in this particular example, non-zero.

13. As per Claim 13, Pilat teaches: The method according to claim 11, comprising a step of:

testing the value of the at least one flag for determining the number of registers containing the data to be saved (Column 5, Lines 5-13, the mode of operation must be tested to determine how much data to save).

14. As per Claim 14, Pilat teaches: The method according to claim 11, comprising the following steps:

restoring the register containing the at least one flag (Column 4, Lines 38-39, a return must return all previously saved states); then

restoring contextual data contained in a variable number of registers that varies according to the value of the at least one flag present in the restored register (Column 5, Lines 29-34).

15. As per Claim 15, Christie teaches: The method according to one claim 11 wherein the register containing the at least one flag is saved last and is restored first. (Page 11, Lines 14-20).

16. As per Claim 16, Christie teaches: The method according to claim 11 wherein the at least one flag is formed by at least one bit of a register containing condition code flags (Page 3, Lines 15-19).

17. As per Claim 17, Padmanabhan teaches: A microprocessor comprising:

a memory array having stored therein contextual data (Page 1, Lines 16-17);

a central processing unit coupled to the memory array (Page 1 Line 13);

a plurality of registers associated with the central processing unit (Page 1, Lines 22-23); and

a stack pointer associated with the central processing unit and being arranged for directing contextual data to be stored (Page 2, Lines 19-23), but fails to teach:

a first group of the registers storing contextual data and a second group of the registers not storing contextual data when a flag has a first value and switching to store contextual data also in the second group of registers when the flag switches to a second value, such that the number of registers that store contextual data is variable;

and where the flag is stored in a register to be saved as part of the program contextual data and where the flag is asserted when an extended portion of a program counter has a portion of an extended address;

a stack pointer associated with the central processing unit and being arranged for directing contextual data to be stored in the first group only or in both the second group and the first group, based on the flag value.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers , allowing more operands to be stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used for vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teaches of

saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie alone.

Furthermore, regarding the limitation of the flag changing according to the value of an extended addressing register of a program counter, while Christie uses a signal to actually modify the flag, Examiner believes there is a basis for it being "according to the program counter". If the extended portion of the program counter (which is inherent in computers) is invalid, the machine clearly cannot possibly be in the extended mode, and if it does contain valid data, than the machine must be in extended mode, thus the extended addressing portion of the program counter is an indication of what the flag needs to be, and even if Christie uses a signal to switch the flag, the extended portion of the program counter could also do it, based on if it contains valid data or not.

18. As per Claim 18, Christie teaches the microprocessor according to claim 17 wherein the second group of registers includes a register which is used as an extended addressing register when the flag is at a first value (Page 3, Lines 15-19).

19. As per Claim 19, Christie teaches: The microprocessor according to claim 17 wherein the second group of registers includes a single register (Page 3, Lines 6-7).

20. As per Claim 25, Padmanabhan teaches: A microprocessor based system comprising:

a memory array (Page 1, Lines 16-17);

a first amount of contextual data stored in at least one register (registers inherently contain contextual data, also see Page 1, Lines 22-23),

a central processing unit coupled to the memory array and operable upon a program switch to save the first amount of contextual data and, wherein the central processing unit is further operable upon a return from a program switch to restore the first amount of contextual data (Page 2, Lines 10-23), but fails to teach:

the register having at least one flag;

if the flag is asserted, a second amount of the contextual data; and

if the flag is asserted, to subsequently restore the second amount of contextual data.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers, allowing more operands to be stored in fast

memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used for vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teaches of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control

register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie and Padmanabhan alone.

21. As per Claim 26, Christie teaches: The microprocessor based system according to claim 25 further comprising a program counter, wherein the flag is asserted when an extended portion of the program counter contains an extended address value (When the extended portion of the program counter contains valid data, the flag must be asserted to be in 32-bit mode, if the extended portion does not contain valid data, then the flag cannot be asserted because it is in 16-bit mode).

22. As per Claim 27, Christie teaches: The microprocessor based system according to claim 25 further comprising a program counter, wherein the flag is not asserted when an extended portion of the program counter contains a null address (When the extended portion of the program counter contains valid data, the flag must be asserted

to be in 32-bit mode, if the extended portion does not contain valid data, then the flag cannot be asserted because it is in 16-bit mode).

23. Claims 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Christie, in view of Pilat.

24. As per Claim 28, Christie teaches: A method of handling contextual data during program switch operations comprising:

asserting a flag if an extended portion of a program counter is not null (Page 3, Lines 15-19, if there is valid data in the extended portion of the PC, the flag must be set);

copying into a memory during a program switch, a first amount of contextual data including the flag (Page 11 Line 40 – Page 12 Line 2); and

restoring the first amount of contextual data during a return from the program switch (Page 12, Lines 1-2), but fails to teach:

if the flag is asserted, copying a second amount of contextual data; and

if the flag is asserted, restoring the second amount of contextual data.

While Christie teaches asserting flags and saving contextual data, Christie saves all registers, and does not save a variable amount of data based on the flag. However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence,

Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teaches of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie.

25. As per Claim 29, Christie teaches: The method according to claim 28 wherein the flag is asserted when the extended portion of the program counter contains an extended address value (Page 3, Lines 15-19, if there is valid data in the extended portion of the PC, the flag must be set).

26. As per Claim 30, Christie and Pilat teach: The method according to claim 28 further comprising updating a stack pointer by a first amount if when a first amount of contextual data is copied and updating the stack pointer by a second amount if the second amount of contextual data is copied (Christie, Page 11, Line 15: data can be put on the stack, and it should be obvious that the stack pointer needs to be adjusted by the appropriate amount depending upon the data to be stored).

Response to Arguments

27. Applicant's first argument is that none of the cited references discuss an extended addressing register, more specifically, an extended addressing register of a program counter, and changing the value of the flag according to the content of an extended addressing register of the program counter. However, in Christie, the extended addressing registers are the extended portions of the registers, the 16 bits which were not present in the 16 bit versions of the registers. A program counter is inherent in computers, and if the computer is operating in 32-bit mode, then the program counter must be 32 bits as well. As for changing the flag according to the content of the extended portion of the program counter, if the value in those 16 bits is null (or not

valid), then the machine clearly cannot have its flag set because it is operating in 16-bit mode. If there is data in the extended portion of the program counter, then the flag must clearly be set, since 32-bit mode is clearly enabled. In this manner, the flag is set based on the program counter, because based on what is in that part of the program counter will directly affect what the flag is. Whether or not Christie looks at the value of the register, and based on that comparison, sets the flag, Examiner cannot say Christie does so (because there is not enough information), however, the flag is set according to what is in the program counter, because it is an indication of what mode is being used, and Examiner believes there is a difference, perhaps subtle, between those two situations. However, Examiner believes that it would be obvious to do so even if that was not the case, for the reasons laid out above.

Regarding Claims 4 and 12, Examiner has maintained the rejection, because the mere presence of the "when's" implies a degree of being conditional, thus there is the potential for one of the cases to never occur, thus allowing the art to read upon it. Should the limitations in this claim be separated into their own claims, for example, having the case of when the content is 0 being one dependent claim, and the case of when the content is not 0 being another dependent claim, then that particular situation is no longer the case, because there are no longer two options. Therefore, if the limitation of the content of the extended addressing register being 0 (as written in Claim 4) was present in a dependent claim without the other limitation, it would be allowable subject matter, but as long as the other when case exists with it, then the conditional nature of the claim will cause it to be rejected.

The arguments presented above apply equally to Independent Claim 17, and the other dependent claims in general, as the argument for these claims are substantially the same.

Regarding the new claims, Examiner believes that the combination of references as used in the new claims (and the older claims) does teach saving a variable amount of data based on the flag, as there is a motivation for only saving the registers in use, and it is inherent for a system to function properly to restore the exact amount of data which was saved, no more and no less (from a stack). If one was to put 10 values on a stack, then try to remove 15, or 5, then not only would the stack be positioned incorrectly for execution, but contextual data would be potentially lost. Therefore, Examiner believes there is sufficient reason for one of ordinary skill to recognize that it is obvious that if you save a variable amount of data, than that variable amount of data must also be restored. It is true that Christie teaches saving all the registers, Pilat uses different instructions to save variable data, and that Shaylor uses dirty bits, however, the Examiner is not bodily incorporating these references, but that together, realizing that they provide a motivation and teaching which is to save only the data you need to save, and that saving extraneous data is a waste of resources, and with that teaching, Christie can be obviously modified to read on the claimed invention.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:45-6:15.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert E Fennema
Examiner
Art Unit 2183

RF

RICHARD L. ELLIS
PRIMARY EXAMINER